Vscode配置指南

**作者:向仔州**

这里提前说明一下，vscode创建的工程不允许有中文路径。

**安装windows平台的C/C++编译器**

千万不要去微软官方下载

微软官方下载完成后就是这么一个软件，点击安装，你会发现在bin目录下面没有gdb调试工具，还要用这个mingw软件去安装其它c/c++编译库。

我还是直接去网上找一个下载好的编译器实在

去这个网站下载<https://sourceforge.net/projects/mingw-w64/files/>

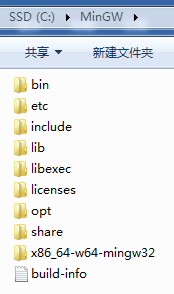
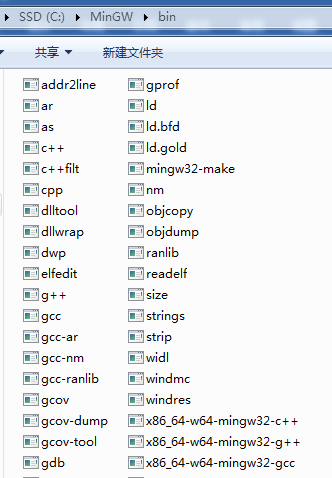
根据你的windows平台来选择，我的电脑是windows64位的，所以我把网页拉到最下方

选择x86\_64-posix-she来下载

下载完成后就是这么一个压缩包

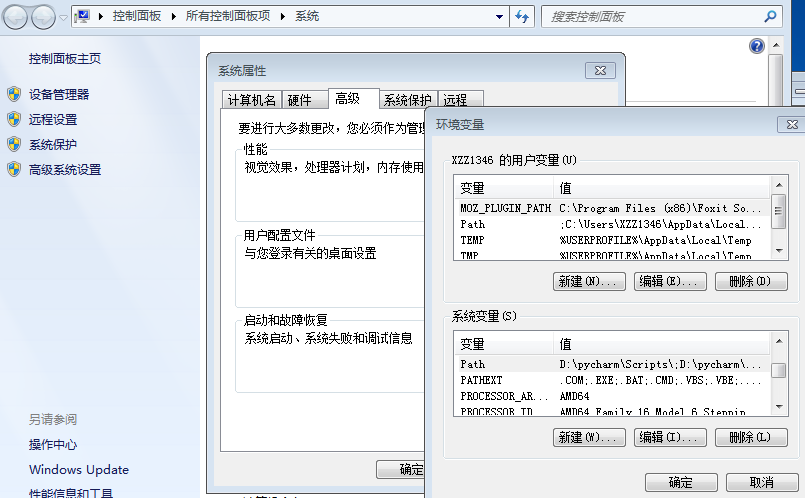
我在C盘创建一个MinGW的目录

将x86\_64-8.1….解压到MinGW目录下

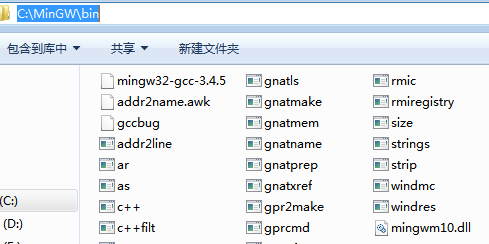
解压后有我要的C/C++库也有gcc/g++/gdb编译器

你看这些交叉编译器都有了，比如gcc g++。

设置环境变量，我在sublime安装手册里面也讲过



设置系统变量下的Path

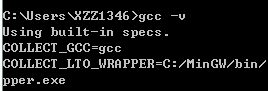


这就是我C/C++交叉编译器所在的路径，主要指定bin目录下的执行文件



在环境变量增加指定的编译器bin目录

执行gcc –v和g++ -v，gdb –v

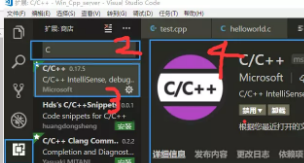
  

这三个工具都显示出版本号，就证明编译器环境变量设置成功。

**VScode使用gcc/g++编译器配置C/C++环境**

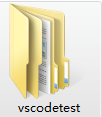
去微软官方下载vscode，然后进行安装



VScode安装好后，打开VScode先安装C/C++语法插件

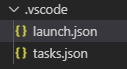
然后建立c++工程，

1.先创建c++工程目录

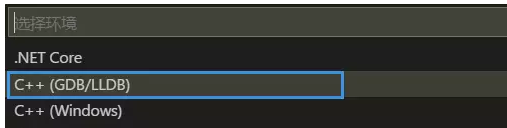
我这里取名为vscodetest

打开Vscode软件，用这个openfolder打开vscodetest工程目录，这样这个目录就是vscode的工程了，你在vscode下面创建的文件都会存放在这个vscodetest目录下，类似sublime工程创建的方式。

然后你需要创建2个配置文件在这个工程里面

这两个目录最好不要自己创建，让工程运行C++程序时候提示你创建

就是这两个配置文件，工程运行后你创建的launch和tasks会自动在.vscode目录下

你运行C++程序后就会出现这个，你点击c++(GDB/LLDB)，就会自动帮你创建launch和tasks。

然后我们只需要修改launch和tasks参数就行了

launch和tasks参数修改

launch文件修改

{

    // Use IntelliSense to learn about possible attributes.

    // Hover to view descriptions of existing attributes.

    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387

    "version": "0.2.0",

    "configurations": [

        {

            "name": "(gdb) Launch",

            "type": "cppdbg",

            "request": "launch",

            "program": "D:\\vscodetest\\build\\test.exe", //这里写入我编译后生产的可执行程序文件路径，因为gdb会去自动寻找这个程序然后执行，你才可以看到输出

            "args": [],

            "stopAtEntry": false,

            "cwd": "${workspaceFolder}", //这里就这样

            "environment": [],

            "externalConsole": true, //这里写true

            "MIMode": "gdb",

            "miDebuggerPath": "C:\\MinGW\\bin\\gdb.exe", //这里填入你的gdb.exe程序位置

            "setupCommands": [

                {

                    "description": "Enable pretty-printing for gdb",//这里一定要改成英文，不然会编译g++ not found file

                    "text": "-enable-pretty-printing",

                    "ignoreFailures": true

                }

            ],

            "preLaunchTask": "compile" //这个名字随便去，但是tasks的label名字必须和这里一样

        }

    ]

}

tasks文件修改

{

    // See https://go.microsoft.com/fwlink/?LinkId=733558

    // for the documentation about the tasks.json format

    "version": "2.0.0",

    "tasks": [

        {

            "type": "shell",

            "label": "compile", //你看label和lunch的preLaunchTask是一样的

            "command": "g++", //这里写你的编译命令，g++或gcc

            "args": [

                "-ggdb",

                "\"${file}\"",

                "--std=c++11",

                "-o",

                "\"${fileDirname}\\${fileBasenameNoExtension}.exe\""

            ],

            "options": {

                "cwd": "C:\\MinGW\\bin" //这里写你的编译器路径

            },

            "problemMatcher": [

                "$gcc"

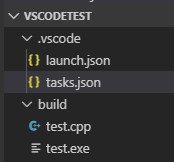
            ],

            "group": "build"

        }

    ]

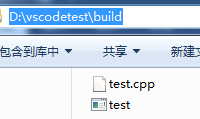
}

你自己建立build目录创建的你的C++程序文件，然后调试就是。

如果运行出现collect2.exe: error: ld returned 1 exit status ，这是程序被执行了，你关闭vscode

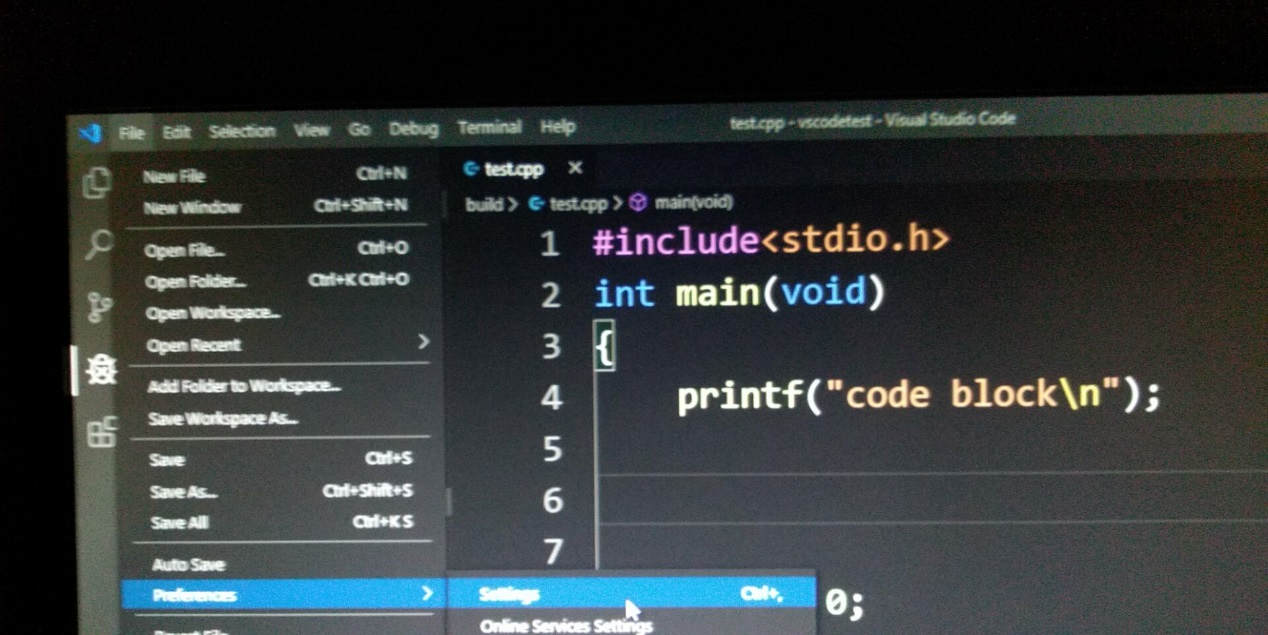
然后看看任务管理有没有你的调试程序进程，有的话就直接关闭，重启vscode就好了

这里有个关键问题一定要记住，vscode的所有工程不允许有中文路径，比如D:\我的桌面\...,这些是不允许的，必须在英文路径下。不然会出现运行报错，说找不到文件什么的。

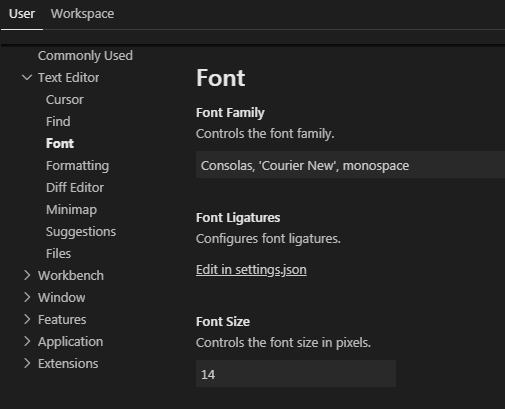
必须是因为路径，这样才正确。

你配置好的launch和tasks在其它新创建的工程里面可以直接复制过去用。放在.vscode目录下就是

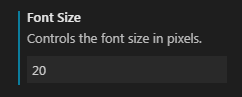
**VScode代码字体大小修改**

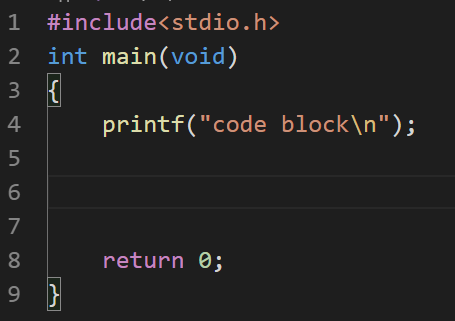


选择preferences->settings

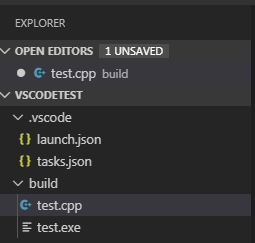


默认代码字体大小是14，我改成20试试。

改完之后一定要执行file->Save All保存

你看字体明显变大了，下次打开VScode也是这样大的字体

**图标插件，让你的目录图标像目录，代码文件图标像代码，看起直观**

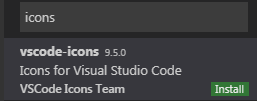


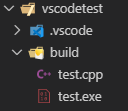
这个符号代表c++文件有点看得出来

这个符号代表json文件，我也看不出来

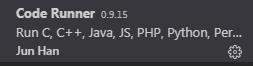
这个符号代表目录，你儿才看得出来

下面安装vscode-icons插件，解决图标难看问题

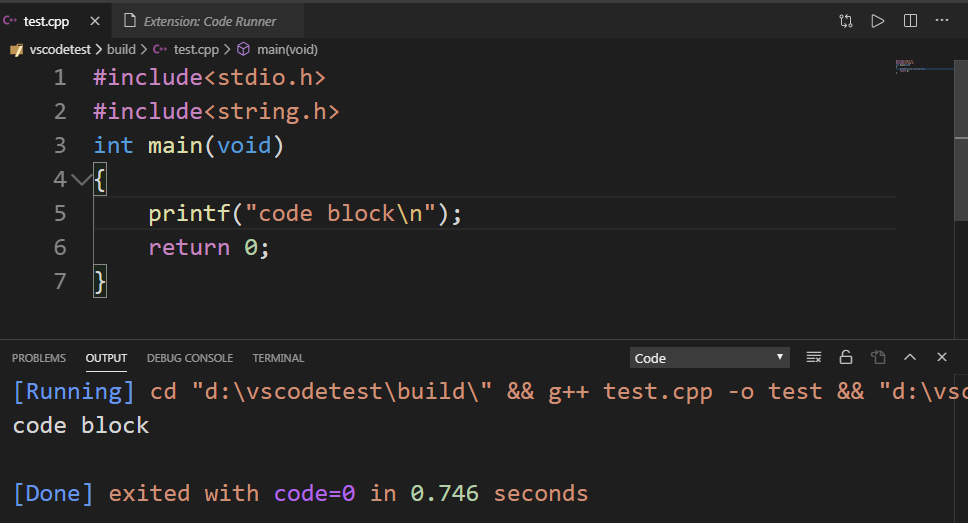


安装后，你看图标一目了然了目录是目录图标代码是代码图标

**代码快速运行插件Code Runner，方便点击运行按钮**

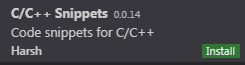


只需要点击这里

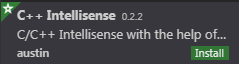


代码在终端运行打印了

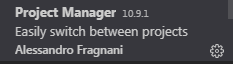
**代码片段C/C++ Snippets插件安装**

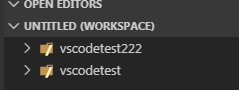
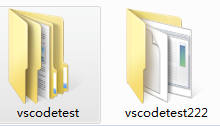


**C++插件intellisense安装**

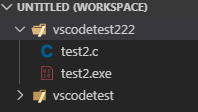


**VScode工程管理插件Project Manager**

安装插件

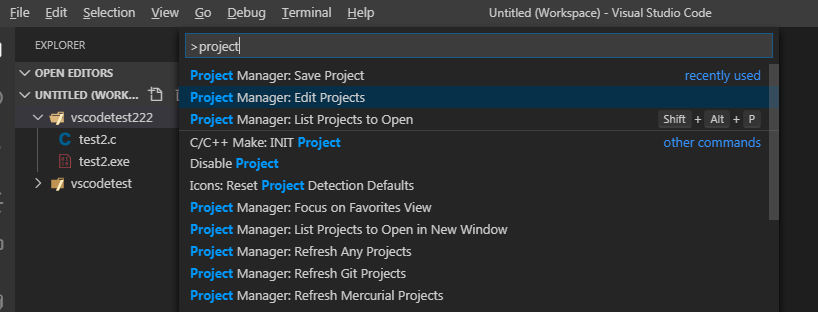
 

这两个目录就等于两个工程



你点击其中一个工程的顶层目录

键盘输入ctrl+shift+p



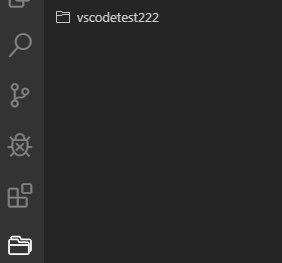
然后输入Project Manager 选择Edit Projects

将产生一个新的json文件



会自动生成工程目录的代码

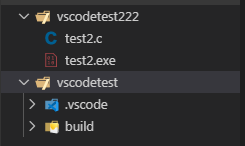
这个就是新json文件



2.这里将出现工程管理文件目录

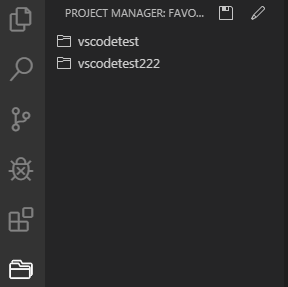
1.点击目录包

如果你想加入新的工程目录，就只需要修改json文件就行

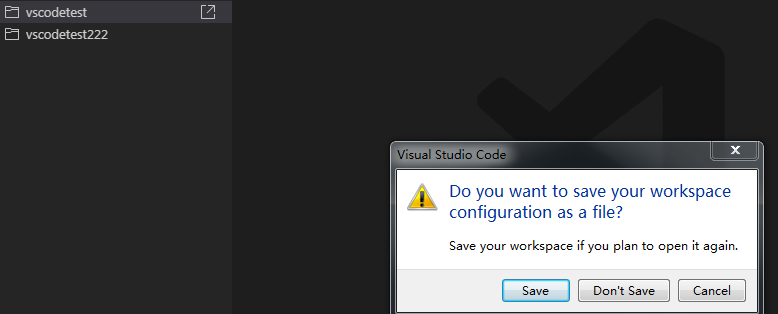
写入新工程名称和工程所在的目录路径

这里加入逗号，然后复制上面的代码

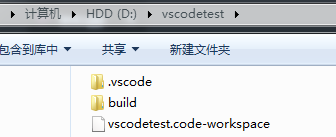


2.包含了两个工程目录，你点击哪个工程，vscode就会打开哪个工程

1.在工程目录包下

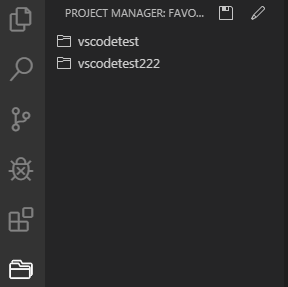


你打开任何一个工程目录，要求你保存工程配置文件，其实就是叫你保存一个配置，这个文件记录了你加入到project Manager里的工程目录有多少

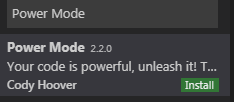


我取了名字，保存在自己安排的目录下

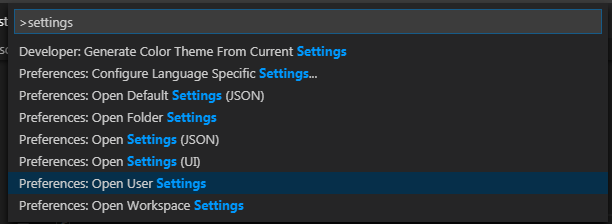
下次从新打开vscode

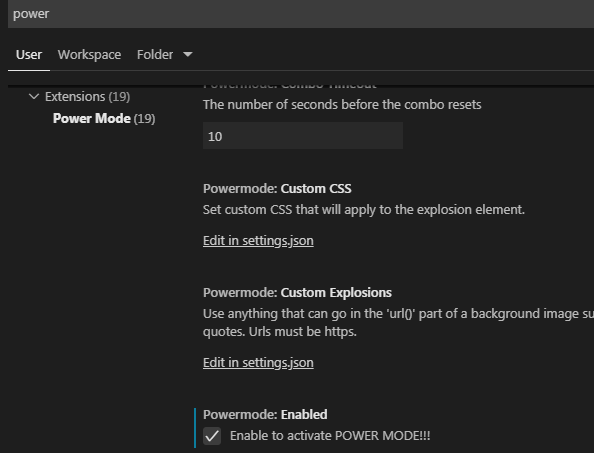
这两个工程都还在工程目录包下，你随便打开就是。

**编辑代码花哨显示插件Power Mode**

安装插件

然后ctrl+shift+p，输入settings

选择open User Settings



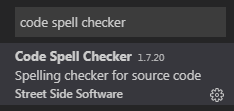
点击使能，这样重启vscode，代码才会有反应

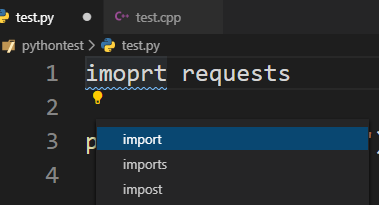
搜索power

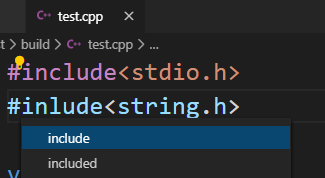
写代码就会有冒泡的效果

**代码关键字写错，提示正确关键字插件code spell Checker**

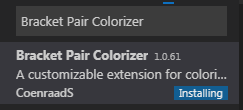
支持很多语言，C/C++,python,Java….等等

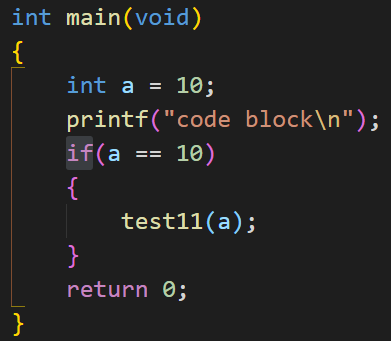
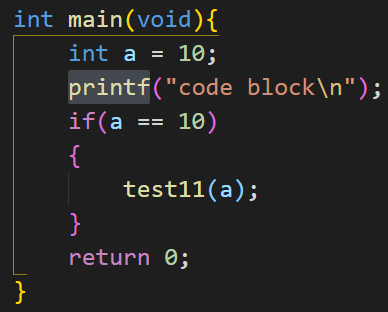


python import写错提示选择

C++ include写错提示选择

**函数代码段颜色更加明显Bracket Pair Colorizer插件**

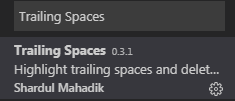


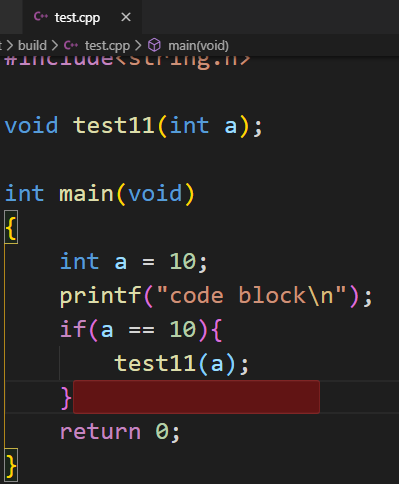
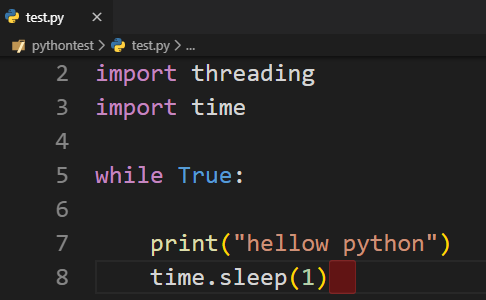
 

说白了就是大括号提示线

你看代码段增加了包括线

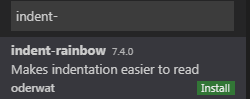
**高亮冗余的空格插件Trailing Spaces，我觉得对python有点用**

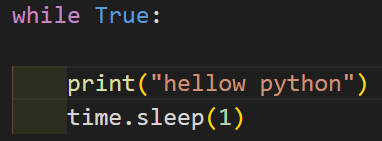


这是C/C++空格检查 这是python空格检查

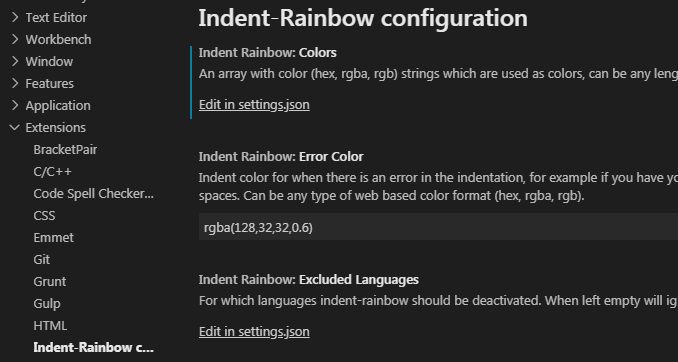
**彩虹缩进插件indent-rainbow，比较适合python**



你看如果是缩进就是这种黄颜色，如果是空格就是红色。

彩虹缩进语法规则

点击file->prefencens->setting



在这里面增加代码，修改tab颜色

输入下面这段语句

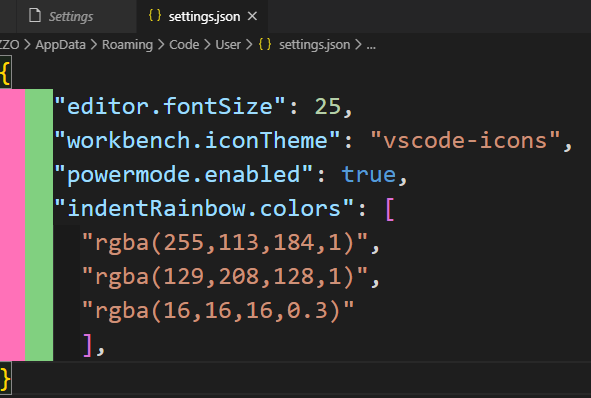
"indentRainbow.colors": [

      "rgba(255,113,184,1)",

      "rgba(129,208,128,1)",

      "rgba(16,16,16,0.3)"

      ],

以此类推

第3个tab颜色

第2个tab颜色

第1个tab颜色

格式就是rgba(红，绿，蓝，透明度) 透明度是0~1,1是不透明，0是完全透明，0.5是半透明，也可以是0.1,0.2,0.3…决定了透明强度。用windows画板决定调色值。

设置完成后重启Vscode就能看见了。

"indentRainbow.colors": [

      "rgba(122,118,105,0.1)",

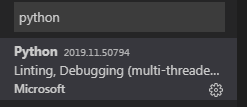
      "rgba(71,67,65,0.1)",

      "rgba(192,192,192,0.1)"

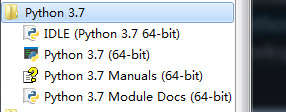
      ],

这是我的配色方案

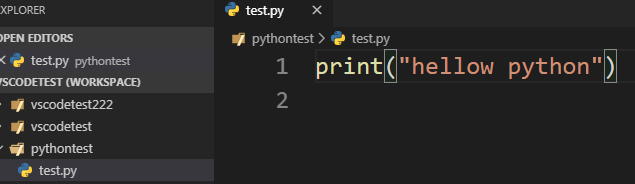
**VScode搭建python环境**

python语法调试器安装

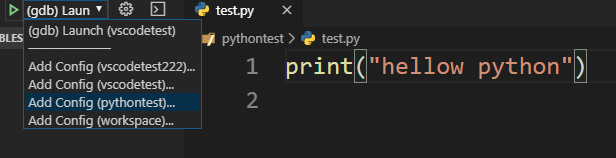
按照另外一份sublime文档把python解释器安装好

我这里是python3.7

插件和解释器安装完成后建立python工程

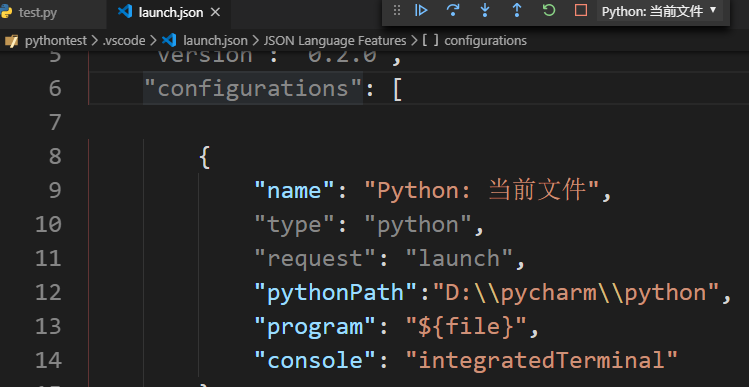


创建python目录和加载目录创建python工程

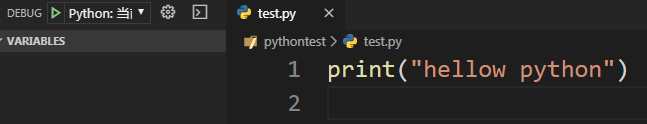


运行python会出现config选择，你选择你自己定义的pythontest工程

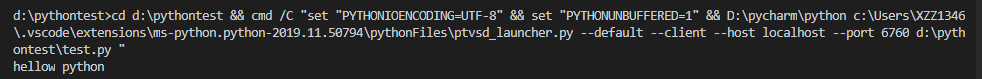
然后点击file，会自动生成launch.json文件



在这里添加pythonPath参数，把你的python解释器路径加入进去就行了

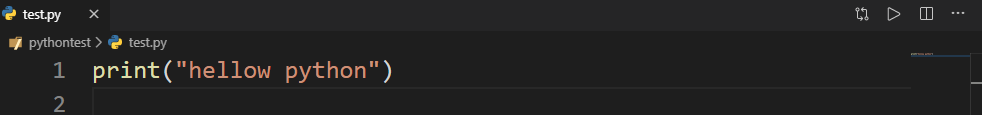


点击这里运行python

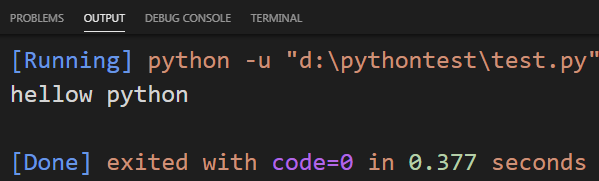


你看程序打印出来了，但是我感觉这个终端不好看

换个运行python的点击方式

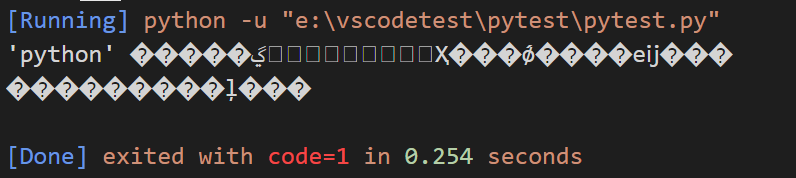


点击内置的python运行按钮

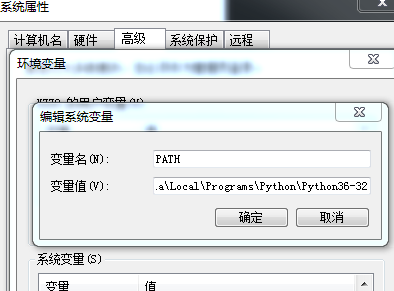


你看这种打印出来就很好看

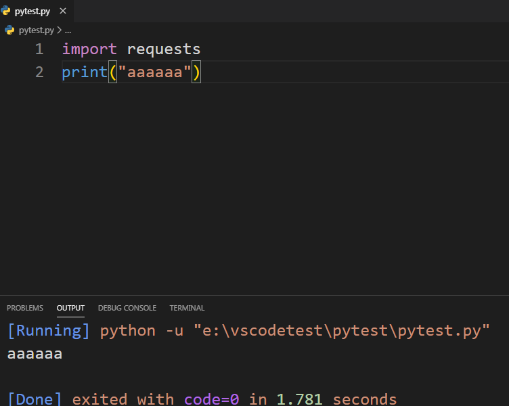
**Python环境搭建好之后点击Run code有些机器会出现错误**



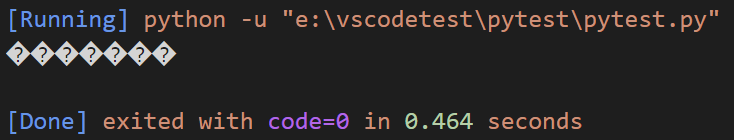
这是因为你忘记在windows电脑加入python的环境变量了



在环境变量中，加入python解释器路径

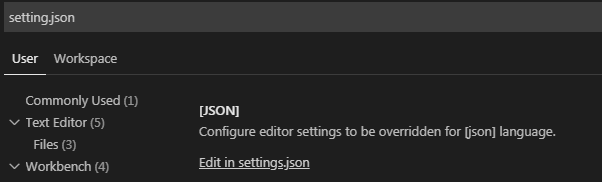
重启VScode，问题得到解决

**Run code运行python程序，遇到中文输出乱码**



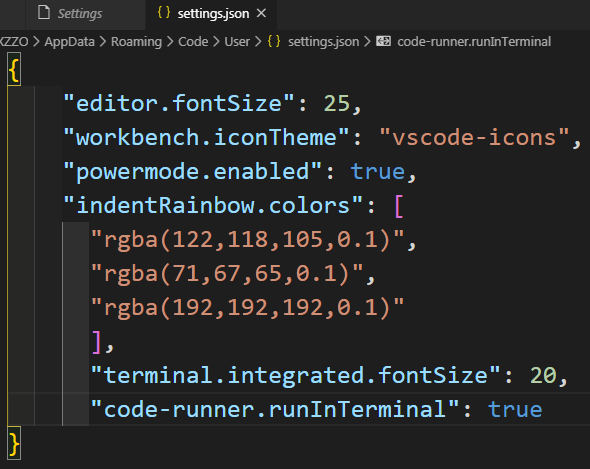
这种情况做好把Run code程序映射到TERMINAL终端运行

打开file->prefercens->settings



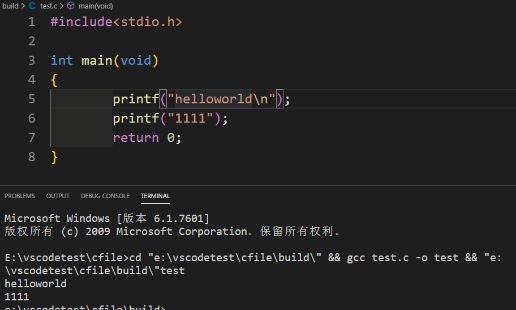
修改setting.json

搜索setting.json



 加入"code-runner.runInTerminal": true

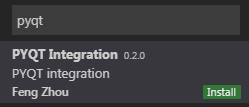
这样python和C++的代码Run code运行都映射到TERMINAL终端了。

C++运行在TERMINAL输出 Python运行在TERMINAL输出

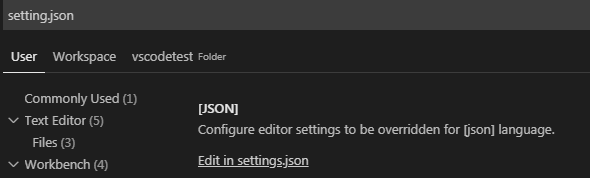
这样主要解决输出显示中文乱码问题

**VScode搭建pyQT5环境**

安装pyqt插件

然后按照pyQT5使用手册在windows下用pip 安装pyQT5库到python解释器目录下。

**编写pyQT5代码出现红色波浪线问题解决，主要是pylint是检查C/C++的stdlib库，所以对python导入的库会报错。**



打开setting.json文件

关闭pylint报错

"python.linting.pylintArgs": [

        "--disable=W,C,E"

      ], //忽略指定报错

     "python.linting.enabled": false //关闭pylint 报错 （不包括语法错误）

问题得到解决。如果你不想这样做也可以在settings.json后面加这句

"python.linting.pylintArgs": ["--generate-members"]

这样也是可以的

Python智能自动补全

在setting.json中加入你pip安装在python解释器目录下的第三方库路径

"python.autoComplete.extraPaths": [

      　　　　"C:/Users/XZZO/AppData/Local/Programs/Python/Python36-32/Lib/site-packages",

      　　　　"C:/Users/XZZO/AppData/Local/Programs/Python/Python36-32/Scripts",

      ],

支持一些python语法格式，这里要安装flake8

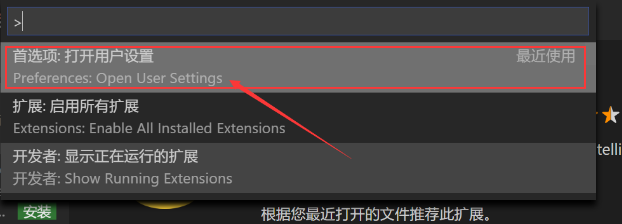
"python.formatting.provider": "yapf",

      "python.linting.flake8Args": ["--max-line-length=10"],

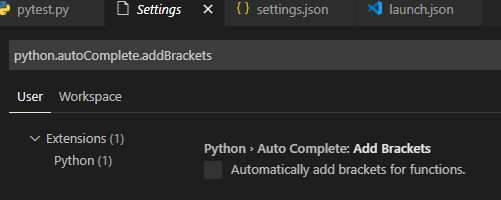
      "python.linting.pylintEnabled": false,

**VScode 的python没有自动补全功能，导致安装的PyQT5也没有自动补全，如何增加自动补全**

使用快捷键Ctrl+Shift+p



在打开的Open User Settings文件界面，搜索：python.autoComplete.addBrackets



勾选上

